UC Usabilidade, Desenvolvimento Web, Mobile e Jogos

A3 UNIFACS LAPA

Equipe:

Rilton Bispo dos Santos: 1272218589

João Vitor Campos Gottschalk: 12722130690 Lucas Davi Barros dos Santos: 1272218717

Douglas Dauto Bastos: 12722133147 Vitor Messias Motta Santos: 12723133980

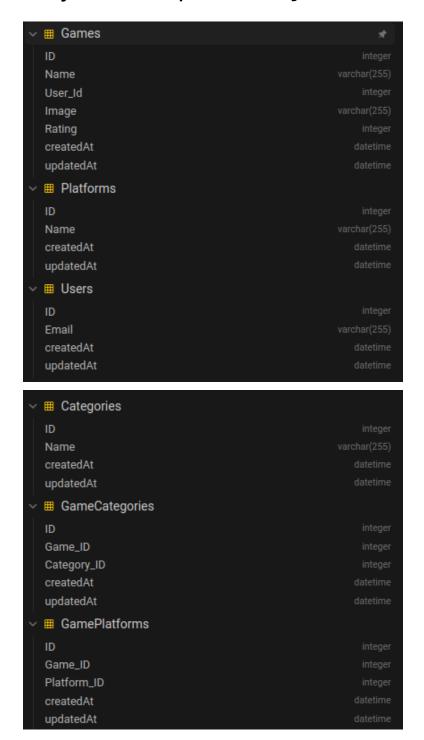
Introdução

O Projeto Catálogo de Jogos, parte da disciplina de Usabilidade, Desenvolvimento Web, Mobile e Jogos na UNIFACS LAPA, tem como objetivo criar uma aplicação WEB para gerenciar o catálogo de jogos de usuários. Este projeto visa proporcionar uma experiência eficiente e organizada na gestão de jogos, promovendo o cadastro de usuários, login, CRUD de jogos e plataformas, categorização, avaliação.

Fundamentação Teórica

Abordamos as tecnologias essenciais para o desenvolvimento do projeto. Node.js foi escolhido para o backend devido à sua eficiência e escalabilidade,para banco de dados usamos SQLite, enquanto React.js proporcionou uma interface de usuário dinâmica e responsiva. Material UI e Bootstrap foram utilizados para o design, garantindo uma experiência visual consistente. O uso do Axios e do Sequelize facilitou a comunicação entre o frontend e o backend, além de otimizar a manipulação de dados no banco de dados (usamos Sequelize apenas para criação das tabelas). O Multer foi empregado para permitir o envio de imagens do jogo.

Projeto de implementação



Tabelas no Banco de dados

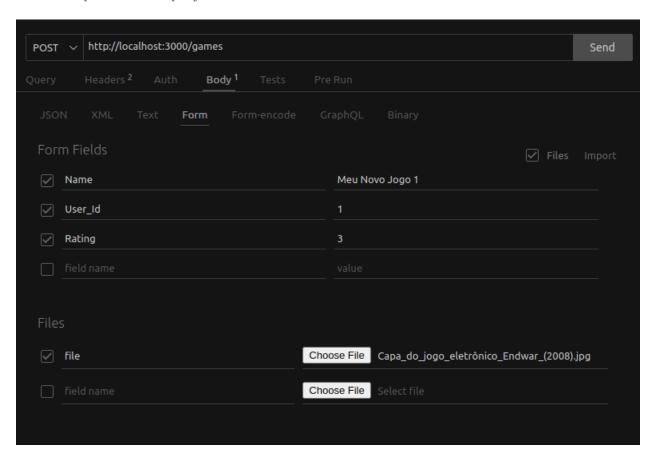
Assim que o projeto back-end é iniciado, ele verifica criar o banco de dados com as tabelas predefinidas já povoadas com alguns exemplos de dados. Para esse projeto é necessário ter a versão v20.10.0 do NodeJs e a versão 10.2.3 do npm.



Comando para rodar o projeto back-end



Comando para rodar o projeto front-end



Print Criação de um Jogo

Para criar um jogo usamos o body do tipo Form-Data

Considerações Finais

Algumas funcionalidades não puderam ser totalmente implementadas. No front end, limitações foram encontradas na atualização de imagem, categoria e plataforma de jogos, sendo possível apenas a modificação do nome e nota, também não conseguimos fazer um sistema de login que verifica qual usuário está logado. No back-end, a utilização de Query para criar registros no banco de dados foi um desafio, uma vez que era um requisito evitar o uso de funções prontas do Sequelize. Contudo, foram utilizadas funções predefinidas apenas para criar, para deletar, atualizar e obter dados foi usado Query.

Para facilitar as requisições HTTP temos uma coleção completa com todos os exemplos que requisições possíveis, para usar-lá baixe a extensão do vscode Thunder Client (rangav.vscode-thunder-client) e faça a importação deste arquivo json: https://drive.google.com/file/d/1Qvka75lkXvhwlowzWme7SUG4fFGsblw6/view?usp=sharing

Link de apresentação do projeto:

https://www.loom.com/share/f3980e42bc544d328dd22563823b79d4?sid=3f6aa1a4-2f02-46c1-a 953-2407ba81a418

Link do repositorio do github: https://github.com/riltonbispo/A3-Final